



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Improving the Capabilities of a Continuum Laser Plasma Interaction Code

J. A. F. Hittinger, M. R. Dorr

June 16, 2006

Scientific Discovery Through Advanced Computing 2006
Denver, CO, United States
June 25, 2006 through June 29, 2006

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

Improving the Capabilities of a Continuum Laser Plasma Interaction Code

J A F Hittinger and M R Dorr

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, L-561,
P.O. Box 808, Livermore, CA 94551, USA

E-mail: hittinger1@llnl.gov

Abstract. The numerical simulation of plasmas is a critical tool for inertial confinement fusion (ICF). We have been working to improve the predictive capability of a continuum laser plasma interaction code **pF3d**, which couples a continuum hydrodynamic model of an unmagnetized plasma to paraxial wave equations modeling the laser light. Advanced numerical techniques such as local mesh refinement, multigrid, and multifluid Godunov methods have been adapted and applied to nonlinear heat conduction and to multifluid plasma models. We describe these algorithms and briefly demonstrate their capabilities.

1. Introduction

In laser-induced inertial confinement fusion (ICF), laser irradiation is used to implode a fuel capsule, generating the high energy density required to initiate fusion. The laser irradiation creates a high temperature plasma that interacts in nonlinear ways with the laser light. In addition to refraction by density gradients, a variety of parametric instabilities (parametric in the light amplitude) exist that convert laser energy into internal plasma waves and scattered electromagnetic waves. Since the irradiation of the fuel pellet requires symmetry for proper implosion and since stray laser light can damage optical systems, understanding the laser plasma interaction (LPI) is of critical importance in ICF experiments.

In support of the National Ignition Facility (NIF) [1], a parallel LPI code **pF3d** [2] was developed by physicists to help understand the complex phenomena involved. This code couples a nonlinear fluid model of the plasma with a paraxial model of the laser light and other stimulated electromagnetic waves. For full simulation of ignition-scale geometries and times, it is impractical to use traditional Particle-In-Cell (PIC) methods. However, a variety of reduced models resulting from PIC simulations on more practicable space-time “patches” are implemented in **pF3d** to approximate non-equilibrium kinetic effects.

In previous work [3, 4], we have applied adaptive mesh refinement and other advanced numerical techniques to laser plasma interaction models similar to those used in **pF3d**. Drawing on the knowledge gained in these investigations, we have been working with physicists to improve the physical and numerical models used in their code. In this paper, we will describe the nonlinear, local electron thermal transport model and the nonlinear, multifluid model that we have developed to extend the capabilities of **pF3d**, and we will show some results of this work relevant to the programmatic work in support of NIF.

2. Laser Plasma Model

A fluid plasma model consists of a system of mass, momentum, and energy equations for each electron and ion species, coupled through a Lorentz force term to Maxwell's equations. However, suitable approximations lead to a much simpler hydrodynamic model in which bulk motion is due to ion motion and only the internal energies of the ion and electron species must be considered independently. For clarity and brevity, we present a simplified form of the governing equations used in the **pF3d** code. Details of the full model can be found in [2, 5].

In our fluid model, the plasma state is completely described by six primitive variables: an ion mass density $\rho(\mathbf{x}, t)$, the hydrodynamic velocity vector $\mathbf{v}(\mathbf{x}, t)$, and the ion and electron pressures $p_i(\mathbf{x}, t)$ and $p_e(\mathbf{x}, t)$, respectively. The plasma system consists of five equations expressing mass, momentum, and energy balances coupled to an evolution equation for one of the internal energies:

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (1a)$$

$$\partial_t (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) + \nabla p = \mathbf{f}_p, \quad (1b)$$

$$\partial_t \left(\frac{3}{2} p + \frac{1}{2} \rho_i |\mathbf{v}|^2 \right) + \nabla \cdot \left[\left(\frac{5}{2} p + \frac{1}{2} \rho_i |\mathbf{v}|^2 \right) \mathbf{v} \right] = \mathbf{v} \cdot \mathbf{f}_p - \nabla \cdot \mathbf{q} + Q, \quad (1c)$$

$$3/2 [\partial_t p_i + \nabla \cdot (p_i \mathbf{v})] + p_i \nabla \cdot \mathbf{v} = 0, \quad (1d)$$

The ion number density is related to the ion mass density by the ion mass, $\rho = m_i n_i$, and by quasi-neutrality, the electron number density is $n_e = Z n_i$, where Z is the charge state. The total pressure, by Dalton's law, is $p = p_i + p_e$, and the ideal gas law relates the pressures to the temperatures, for example, $p_e = n_e T_e$. We note that, in ICF LPI applications, $0 < p_i/p \ll 1$, so numerically it is better conditioned to solve for p_i instead of p_e .

For unbounded domains spanning thousands of light wavelengths, numerical integration of the Maxwell or Helmholtz equations is exceedingly expensive. Instead, we model light waves using one-way paraxial wave equations, which are an approximate factorization of the wave equation along a dominant direction of propagation [6]. For example, for light propagating primarily in the z -direction, we assume a time-harmonic vector potential $A(\mathbf{x}, t) = \text{Re}\{\hat{A}(\mathbf{x}, t) \exp(-i\omega t + i\varphi)\}$ with a local phase factor $\varphi(z, t) = k \int_{z_0}^z \eta_0(\zeta, t) d\zeta$, where ω and k are the light frequency and wavenumber, respectively, $\hat{A}(\mathbf{x}, t)$ is a complex-valued envelope, $\eta_0(z, t) = \sqrt{1 - \bar{n}_e(z, t)/n_c} > 0$ is a transverse-averaged refractive index, and n_c is the critical density of the plasma. In this case, the paraxial equation has the pseudo-differential form

$$\left(\partial_t + c \eta_0 \partial_z - \frac{ic \eta_0 \nabla_\perp^2}{k \eta_0 + \sqrt{k^2 \eta_0^2 + \nabla_\perp^2}} + \nu + i \partial_t \varphi + \frac{c}{2} \partial_z \eta_0 \right) \hat{A} = -i\alpha \delta n \hat{A}, \quad (2)$$

where $c = \omega/k$ is the speed of light, $\delta n(\mathbf{x}, t) = n_e(\mathbf{x}, t) - \bar{n}_e(z, t)$, $\nu(n_e, T_e)$ is the collisional absorption rate, and α is a real, constant factor. It should be noted that the integrating factor φ reduces the magnitude of the phase variation in the resulting envelope, thereby reducing the computational cost. The **pF3d** code uses Fourier spectral methods to advance several coupled paraxial equations of the form (2) that model the incident laser as well as stimulated light and plasma waves [5].

We have included only terms on the right-hand side of (1) that are relevant for this discussion. The electron heat conduction is represented symbolically by $-\nabla \cdot \mathbf{q}$; in the Spitzer-Härm approximation, the heat conduction explicitly takes this divergence form with a nonlinear heat flux vector $\mathbf{q}(n_e, T_e)$. The ponderomotive force $\mathbf{f}_p \propto -n_e \nabla |\hat{A}|^2$ in (1b) represents a “light pressure” exerted on the plasma through the fast oscillation of electrons in the light waves. Similarly, the absorption term $Q \propto \nu |\hat{A}|^2$ in (1c) represents the heating of the electrons by inverse Bremsstrahlung absorption of the laser light.

In the **pF3d** code, the coupled, nonlinear system (1) and (2) is advanced explicitly through a sequence of operator-split steps. In the basic order, the light is advanced, followed by electron heat conduction, and then the remaining hydrodynamic terms are advanced. Typically, the light and heat conduction steps are sub-cycled together because of their tighter coupling and faster physics. This process formally should converge with at best a first-order rate in time. Working to improve algorithms within this framework, we have sought to develop algorithms which independently achieve second-order, unless doing so would incur significant additional expense.

3. Two-Level, Multigrid Spitzer-Härm Electron Heat Conduction

Originally, the **pF3d** LPI code was developed to study parametric instabilities due to wavelength-scale structure within smoothed laser beams. Beam conditioning, such as smoothing, reduces spatial coherence in the beam but promotes better beam propagation in plasma for high intensity lasers because it reduces beam spray due to self-focusing. Over time, **pF3d** has been applied to problems of increasing size, scaling up from tens to tens of thousands of wavelengths. While periodic transverse domains and a nonlocal, linear electron heat transfer model [7] were sufficient for earlier studies, models more applicable to larger-scale problems were needed.

In response, we have developed a local, nonlinear Spitzer-Härm electron heat conduction model for the **pF3d** code. This model makes use of a two-level, static mesh refinement that provides an extended domain on which heat, deposited through absorption, is conducted. The two-level composite solution is advanced through a Fast Adaptive Composite (FAC) algorithm that uses a scalable, geometric multigrid algorithm as the solver.

3.1. Spitzer-Härm Model

In the electron heat conduction step, the governing equation is

$$\frac{3}{2}n_e \frac{\partial T_e}{\partial t} = \nabla \cdot \mathbf{q} + Q. \quad (3)$$

The advective and pressure work terms as well as the laser propagation are handled in a separate step, thus n_e and Q do not vary in time for this temperature update step. The Spitzer-Härm heat flux [8] has the form $\mathbf{q} = -\kappa \nabla T_e$, where $\kappa = \beta T_e^{5/2} / \ln(\Lambda)$ is the classical, nonlinear Spitzer-Härm thermal conductivity. The symbol $\ln(\Lambda)$ symbolically represents the contribution of temperature- and density-dependent electron-ion and electron-electron Coulomb Logarithms, and β is a material-dependent constant.

Local Spitzer-Härm heat transfer models the electron energy transport as a diffusion process, which is problematic when the heat flux is greater than its physical limit. We employ a standard flux limiter to constrain the transport, limiting the magnitude of the heat flux to a fraction of the local free-streaming electron energy flux, $n_e T_e v_e$.

3.2. Discretization

Consider a three-dimensional, uniform grid with cells of size $\Delta x \times \Delta y \times \Delta z$. Let the grid cell centers be indexed by 3-tuples (i, j, k) , and let the cell faces be indexed by $(i \pm 1/2, j, k)$, $(i, j \pm 1/2, k)$ and $(i, j, k \pm 1/2)$, in the x , y and z directions, respectively. We advance over a time $\Delta t \equiv t^{n+1} - t^n$ and denote by $T \equiv \{\bar{T}_{i,j,k}^{n+1}\}$ the vector of unknown temperatures. Integrating (3) over cell (i, j, k) and using a backward Euler temporal differencing, the discrete system for the cell-averaged electron temperature can be written as

$$\begin{aligned} \frac{3}{2} \bar{n}_{i,j,k} \frac{\bar{T}_{i,j,k}^{n+1} - \bar{T}_{i,j,k}^n}{\Delta t} + \frac{(q_x)_{i+1/2,j,k}^{n+1} - (q_x)_{i-1/2,j,k}^{n+1}}{\Delta x} \\ + \frac{(q_y)_{i,j+1/2,k}^{n+1} - (q_y)_{i,j-1/2,k}^{n+1}}{\Delta y} + \frac{(q_z)_{i,j,k+1/2}^{n+1} - (q_z)_{i,j,k-1/2}^{n+1}}{\Delta z} - \bar{Q}_{i,j,k} \equiv F(T) = 0, \quad (4) \end{aligned}$$

which is approximate to second order in the spatial mesh size. The backward Euler temporal discretization, while exceedingly stable, is only first-order accurate, but the splitting employed in `pF3d` limits the temporal convergence rate to first-order regardless, so we have opted for the simplicity and efficiency of this method.

The face-averaged heat flux quantities are approximated using averages and differences of the cell-centered quantities at the appropriate time step. For example,

$$(q_x)_{i+1/2,j,k}^{n+1} \equiv -\kappa \left[\frac{n_{i+1,j,k} + n_{i,j,k}}{2}, \frac{T_{i+1,j,k}^{n+1} + T_{i,j,k}^{n+1}}{2} \right] \frac{T_{i+1,j,k}^{n+1} - T_{i,j,k}^n}{\Delta x}. \quad (5)$$

In practice, considerable computation time can be saved, without loss of accuracy, if the Coulomb logarithms in the conductivity are computed at cell centers and averaged to the interfaces. This reduces by a factor of three the number of Coulomb logarithm calculations, each of which involves expensive transcendental and algebraic function evaluations.

By Newton's method, the solution of (4) is approximated by a sequence of iterates obtained from a linearization of F :

$$T^{(m+1)} \equiv T^{(m)} + \delta T^{(m)}, \quad (6)$$

where the update $\delta T^{(m)}$ is the solution of the linear system

$$\partial_T F(T^{(m)}) \delta T^{(m)} = -F(T^{(m)}). \quad (7)$$

The exact Jacobian $\partial_T F$ is not essential for the convergence of the Newton iteration, so we approximate the Jacobian by neglecting terms involving temperature derivatives of the conductivity. Such approximation does not affect the accuracy of the nonlinear solution, only (possibly) the convergence rate of (6). For the system (4), this approximate Jacobian matrix $J(T^{(m)})$ has a nine-stripe form corresponding to the centered-differenced diffusion operator.

3.3. Two-Level Mesh Refinement

We define a coarsely gridded *collar* mesh surrounding the standard `pF3d` simulation grid that allows the electron temperature to be computed on a larger domain. This provides the ability to specify the artificial transverse boundary condition in an effective far-field, rather than at the transverse boundaries of the standard `pF3d` mesh. The latter is problematic when the conduction is large enough to cause the boundary temperature to change with time. By using a coarser collar grid, the cost of the two-level approach is mitigated.

Let \mathcal{G}_f denote a standard `pF3d` mesh, and let \mathcal{G}_c denote a coarser, uniform mesh over a larger domain. \mathcal{G}_f is embedded as a locally refined subgrid in \mathcal{G}_c , as shown in Figure 3.3 for a two-dimensional example. Denote by $\mathcal{I}_f^c(\mathcal{G}_f)$ the restriction of \mathcal{G}_f to \mathcal{G}_c ; the collar grid is the complement $\mathcal{G}_c \setminus \mathcal{I}_f^c(\mathcal{G}_f)$. We discretize (3) on the *composite grid* $\mathcal{G}_C \equiv \mathcal{G}_f \cup (\mathcal{G}_c \setminus \mathcal{I}_f^c(\mathcal{G}_f))$.

Since the grids \mathcal{G}_f and \mathcal{G}_c are each uniform, the spatial discretization (4) can be applied in the interiors of these grids. The change in grid resolution at the interface between \mathcal{G}_f and the collar grid requires a nonuniform differencing stencil in the composite grid cells adjacent to the interface. The modified stencils are defined using slave nodes, denoted by 'o' in Figure 3.3, at which values are determined by quadratic interpolation from neighboring coarse grid nodes. The interpolated values provide Dirichlet conditions for the discrete conduction operator on \mathcal{G}_f .

Application of Newton iteration to solve the composite grid analog of the nonlinear system (4) requires the solution of the composite grid Jacobian system

$$J_C(T_C^{(m)}) \delta T_C^{(m)} = -F_C(T_C^{(m)}). \quad (8)$$

To accomplish this, we employ the Fast Adaptive Composite (FAC) algorithm [9, 10], where for each Newton iterate, the linearized system (8) on the composite grid is itself solved iteratively. The procedure for each iteration of the FAC algorithm is:

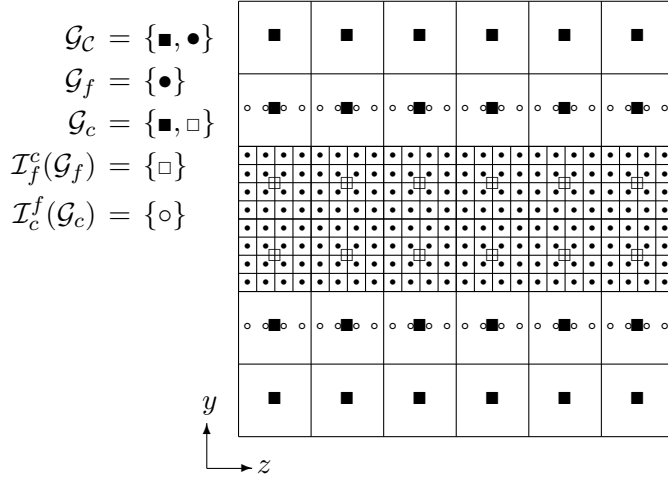


Figure 1. Relationships between grids are shown. The composite grid \mathcal{G}_C is comprised of all of the fine grid cells (\bullet) plus all of the coarse grid cells (\blacksquare) not in the fine grid domain Ω_f . The values at the coarse grid cells in the overlap (\square) are obtained by volume averaging the fine cells contained within each coarse cell (\mathcal{I}_f^c). On the fine grid, the interior boundary conditions are set at slave nodes (\circ) by interpolating nearby coarse cell values (\mathcal{I}_c^f).

- (i) restrict the linear system residual on the composite grid \mathcal{G}_C to the coarse grid \mathcal{G}_c ;
- (ii) solve the linear system on the coarse grid \mathcal{G}_c ;
- (iii) restrict the linear system residual on the composite grid \mathcal{G}_C , with corrections on the fine grid boundary due to the coarse grid solve, to the fine grid \mathcal{G}_f ;
- (iv) solve the linear system on the fine grid \mathcal{G}_f with the Dirichlet conditions at the slave nodes;
- (v) update the composite iterative solution $\delta T_C^{(m,l+1)}$.

This process is continued until the RMS norm of the composite residual is below some tolerance.

3.4. Multigrid

The LMG geometric multigrid algorithm [11], with an adaptive schedule, is used for the linear system solves on \mathcal{G}_f and \mathcal{G}_c . Multigrid algorithms employ a hierarchy of grids, generated by successively coarsening the initial grid by a factor of two in each coordinate direction until a coarsest grid is obtained. On each grid level, a coarsened Jacobian operator is created by averaging coefficients from the next finer level. The grid levels and coarsened operators thereby obtained are used to damp error components at different wavelengths via an iterative process. Starting with the finest grid, one or two *pre-smoothing* iterations of red-black Gauss-Seidel are performed to relax the solution error locally. The resulting residuals are averaged to the next coarser level, where one or more smoothing sweeps are performed to reduce the error components at this coarser resolution. Eventually, the residual is averaged to the coarsest grid, where a direct Gaussian elimination solve is performed. Corrections to the solution are then interpolated from coarser levels to the finer levels, again performing one or more *post-smoothing* sweeps on each level. A single traversal of the hierarchy in this way constitutes one *V-cycle*. We have found that two pre- and post-smoothing sweeps per level is sufficient in application.

It can be proved that, under conditions satisfied in our implementation, the number of V-cycles required to achieve a fixed level of accuracy is asymptotically independent of the mesh size. This implies that the algorithm will scale well as the problem size is increased along with the number of processors, since the dominant cost is the scalable smoothing sweeps.

Since the composite Jacobian system (8) is solved within a Newton iteration, it usually makes little sense to set tight linear solver tolerances (*i.e.*, small in value), since this wastes effort if the Newton iteration has not converged. For this Spitzer-Härm problem, we have found that a better strategy is to set the linear solve tolerance as large as possible, thereby reducing the cost of the multigrid solve, while still converging the outer Newton iteration in several iterations.

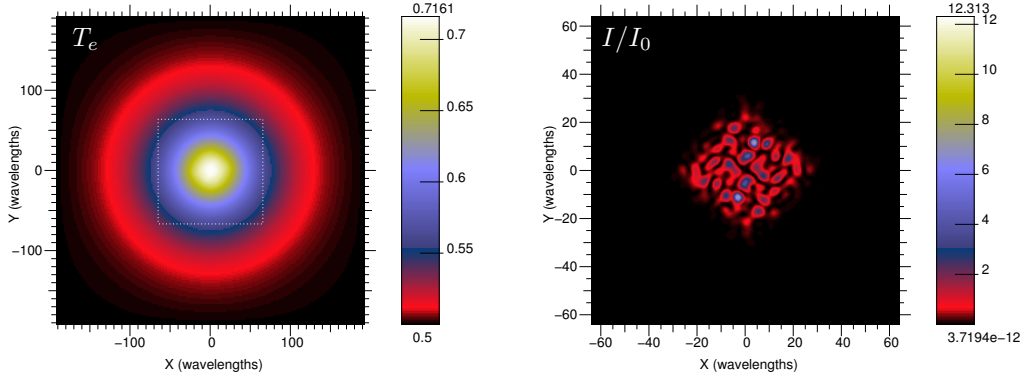


Figure 2. Spitzer-Härm results at $t = 2.89$ ps at $z = L_z/2 = 128\lambda$ for a fine mesh resolution of $(n_x, n_y, n_z) = (256, 256, 64)$, fine-to-coarse refinement ratios of 4:1 in each dimension, and a coarse mesh spatial extent of $3L_x \times 3L_y \times L_z$. Shown are T_e in keV on the composite grid \mathcal{G}_C , with the fine grid boundary outlined, and laser intensity I/I_0 on the fine grid \mathcal{G}_f , $I_0 = 2(10^{15})$ W/cm².

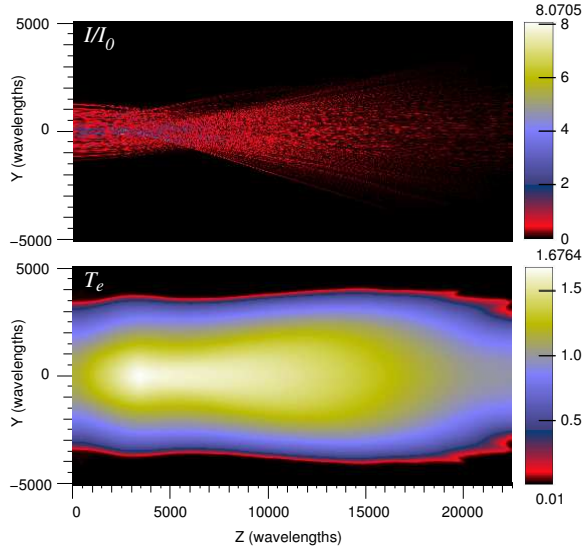


Figure 3. Laser intensity I/I_0 ($I_0 = 2(10^{15})$ W/cm²) and electron temperature T_e in keV for NEL CO₂ gaspipe simulations. Results at $t = 3.5$ ns for a fine mesh resolution of $(n_y, n_z) = (10240, 5632)$, fine-to-coarse refinement ratios of 32:1 in y and z , respectively, and a coarse mesh with twice the spatial extent of the fine mesh in the y direction. The fine mesh simulation domain was $10240\lambda \times 22528\lambda$, and each run took 240 to 360 hours on 64 processors of an Itanium2 Linux cluster.

3.5. Numerical Results

To demonstrate the capability of the two-level Spitzer-Härm algorithm, we present in Figure 2 cross-sections at best-focus of the electron temperature and laser light for a diamond cross section $f/4$ random-phase-plate smoothed beam of wavelength $\lambda = 0.351 \mu\text{m}$. The initial, quiescent plasma conditions are $n_e = 0.06n_c$, $T_e = 500$ eV, and $T_i = 0.5T_e$. For this small spot-size (32λ), the small plasma volume within the beam heats rapidly. Note that the coarse/fine mesh boundary is not visible in the temperature contours. Furthermore, little imprint of the beam structure is visible within temperature field because of the rapid heat transfer rate. The use of a two-level grid allows the temperature on the fine grid boundary to adjust more realistically than would a Dirichlet condition imposed on that same boundary.

A second, NIF-relevant example of the Spitzer-Härm algorithm is shown in Figure 3. These 2D simulations were made in support of a NIF Early Light (NEL) gaspipe experiment to evaluate the effects of beam-smoothing options on the “burn through” of the laser. Cooler plasma absorbs more laser energy, so the burn-through rate is dependent on the details of the plasma heating. Shown are the fine-grid electron temperature and corresponding laser intensities. Due to the large computational domain, full 3D simulations were not feasible, but to account better for the

3D energy loss, a variation of the FAC algorithm was employed, using a 3D coarse grid, and cylindrically rotating the heat source from the 2D grid onto the 3D grid. It became apparent that the temperature was sufficiently smooth that it was unnecessary to compute the heat conduction on the fine grid, so we subsequently solved for T_e on the coarse grid and interpolated this solution up to the fine grid. This procedure resulted in a significant cost reduction of the heat conduction step. The simulation results reproduced the laser transmission trends of the experiments and identified laser filamentation as the primary mechanism responsible for differences between different beam-smoothing techniques.

4. Nonlinear, Multifluid Hydrodynamics

Within the hohlraum target of indirect-drive ICF experiments, the plasma does not have a uniform composition. Distinct regions of low- and high- Z plasmas are generated as the hohlraum and its contents are ionized. To accommodate the simulation needs of the LLNL LPI theorists, we have developed an efficient, nonlinear, multifluid hydrodynamics capability for pF3d. Using volume fractions to track an arbitrary number of fluids, a careful splitting of the physics leads to a nonlinear hyperbolic system that can be solved effectively using a high-resolution Godunov scheme. We discuss the algorithm and present some preliminary simulation results.

4.1. Multifluid Model

We consider a plasma with N distinct material regions. In the hydrodynamic step, which is operationally split from the ponderomotive forcing and electron heat conduction, we simultaneously integrate a nonlinear hyperbolic subsystem of equations, that is, the left-hand side of (1), and augment this with $(N - 1)$ scalar advection equations for volume fractions. These may be written in the conservative form

$$\partial_t (\rho \psi_m) + \nabla \cdot (\rho \psi_m \mathbf{v}) = 0, \quad m = 1, 2, \dots, N - 1, \quad (9)$$

where $\psi_m(\mathbf{x}, t)$ is the volume fraction of the m -th fluid. The N -th volume fraction is computed from the consistency constraint $\sum_{m=1}^N \psi_m = 1$. Locally, mass and ionization state are functionals of the volume fractions, that is, volume averaged by component. For example, $\bar{Z}[\psi_m] = \sum_{m=1}^N \psi_m \bar{Z}_m$, where \bar{Z}_m is the ionization state of fluid m .

An important observation is that, in the absence of forcing and diffusion terms, equations (1a)-(1c) form a closed system *independent* of multifluid effects. This subsystem is, in fact, the Euler equations expressing conservation of mass, momentum, and energy. That the core hydrodynamic behavior can be determined independently from the ψ_m is a consequence of the assumption that all species act as thermally perfect gases with $\gamma = 5/3$. As a result, this plasma fluid formulation solved in divergence form will not suffer from the well-known pressure oscillation problem at material interfaces [12]. The remaining two equations (1d) and (9) are necessary for other steps in the split physics algorithm in order to distinguish between the contributions of electrons and ions to internal energy and to distinguish between different fluid species.

An interesting wrinkle in the solution of (1) is that the ion pressure equation is inherently in nonconservative form, yet the system admits discontinuous solutions. In [3], this equation was discretized in nonconservative form and updated after the conservative update. However, the eigenstructure of the coefficient matrix of the one-dimensional system $\partial_t \mathbf{u} + \mathbf{A} \partial_x \mathbf{u} = 0$ indicates that the ratio p_i/p is preserved across acoustic waves and only changes across a linearly degenerate wave with the flow velocity. Indeed, there is no physical mechanism, such as thermal equilibration, in the hyperbolic subsystem of (1) capable of redistributing the internal energy between electrons and ions, and the eigenanalysis suggests that the pressure ratio p_i/p should obey a scalar advection equation. Formal manipulation of (1d) leads to an equation for the

pressure ratio in divergence form:

$$\partial_t \left(\rho_i \frac{p_i}{p} \right) + \nabla \cdot \left(\rho_i \frac{p_i}{p} \mathbf{v} \right) = 0. \quad (10)$$

With the ion pressure equation in this form, the entire system comprised of the left-hand sides of (1) and (9) is a hyperbolic system of conservation laws.

4.2. Discretization

In [3], we made use of a coupled conservative update that corrects for multidimensional effects [13, 14]. This method requires more computational time and memory than a method based on standard dimensional operator splitting, but produces only marginally better results for most LPI applications. We therefore have opted to use the simpler dimensional split approach, implemented using the standard second-order interleaving of steps [15]. Accordingly, we will describe the hyperbolic advancement algorithm that is applied dimension by dimension.

In one-dimension, the system of conservation laws can be written in the compact form

$$\partial_t \mathbf{u} + \partial_x \mathbf{f} = 0, \quad (11)$$

where $\mathbf{u} = (\rho_i, \rho_i \mathbf{v}, (5p + \rho_i |\mathbf{v}|^2)/2, \rho p_i/p, \rho \psi)^T$ is the state vector of conserved variables, \mathbf{f} is the flux vector, and $\psi \equiv (\psi_1, \psi_2, \dots, \psi_{N-1})$. A finite-volume form of this equation is solved that guarantees convergence to weak solutions (*e.g.*, shock waves). Integrating (11) over the space-time domain $[t^n, t^{n+1}]$ and cell (i, j, k) , one obtains the conservative update

$$\bar{\mathbf{u}}_{ijk}^{n+1} = \bar{\mathbf{u}}_{ijk}^n - \frac{\Delta t}{\Delta x} \left(\hat{\mathbf{f}}_{i+1/2}^{n+1/2} - \hat{\mathbf{f}}_{i-1/2}^{n+1/2} \right)_{jk}, \quad (12)$$

where $\bar{\mathbf{u}}_{ijk}$ represents a cell-averaged state vector and $\hat{\mathbf{f}}_{i\pm 1/2,j,k}^{n+1/2}$ represents time- and face-averaged flux vectors. This expression simply states that the change in the cell average value is due to the average fluxes into and out of the cell. The accuracy of the average fluxes dictates the accuracy of the algorithm, and we approximate these fluxes using the Godunov approach.

The original Godunov method [16] solves Riemann problems, using the piecewise-constant cell averages as initial data, to approximate the flux vector at each cell interface. This naturally and correctly provides a coupled, upwinded discretization for the system. However, the exact Riemann solution is expensive to construct, and much of the resulting detail is unimportant. Instead, approximate Riemann solvers are preferred, and we use the HLLC Riemann solver [17, 18], where the solution to the Riemann problem is modeled as four constant states $[\mathbf{u}_L, \mathbf{u}_L^*, \mathbf{u}_R^*, \mathbf{u}_R]$ separated by three discontinuous waves propagating from the origin with speeds $c_L < c_* < c_R$. Consistency with the governing equation and physical solution provide constraints that determine \mathbf{u}_L^* and \mathbf{u}_R^* , and the extension to our system including the pressure ratio and volume fraction equations is straight-forward. There exists a choice of the signal speeds $\{c_L, c_*, c_R\}$ that guarantees a upwind, positivity-preserving update [19, 20], and we use these speeds to ensure the positivity of density and pressure from the hyperbolic advancement.

The Godunov method using piecewise constant data in each cell is only first-order accurate in space and time. To increase the spatial accuracy, Van Leer [21] devised an approach to reconstruct the solution slope from the cell averages. To preserve monotonicity, these slopes are limited based on the local solution behavior, and the scheme reverts to first-order in the vicinity of discontinuities but achieves second-order in smooth regions.

We adopt this approach and reconstruct on primitive variables $\mathbf{w} = (\rho_i, \mathbf{v}, p, p_i, \psi)^T$, as these are the variables that should not exhibit numerical oscillations. In the x -direction, for cell (i, j, k) , we compute a limited average slope based on the two one-sided differences:

$$\sigma_{ijk}^x = \text{limavg}(\mathbf{w}_{i+1,j,k} - \mathbf{w}_{i,j,k}, \mathbf{w}_{i,j,k} - \mathbf{w}_{i-1,j,k}) / \Delta x, \quad (13)$$

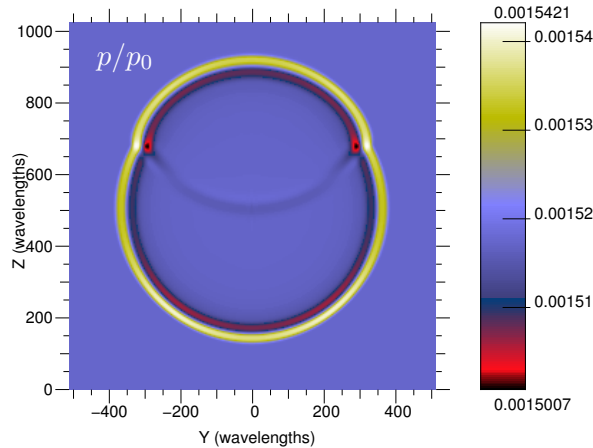


Figure 4. Multifluid calculation demonstrating constancy of total pressure across material interface. Two-dimensional results at $t = 464$ ps on a square domain of length 1024λ ($\lambda = 0.351\mu\text{m}$), using 256 cells in each direction. The initial plasma composition was Au for $z < 512\lambda$ and CH for $z \geq 512\lambda$. The initial conditions were $n_e/n_c = 0.1$, $T_e = 3$ keV, $T_i = Z(\psi)/\max Z$, and $\mathbf{v}/c_{s,\text{CH}} = (0.0, 0.5, 0.0)$ everywhere except at $y^* = z^* = 341\lambda$, where p and T_e were set to ten times the background values p_0 and T_{e0} .

where the average is applied component by component to the primitive state vector \mathbf{w} . We generally use the *double minmod* limiter [21], which chooses the smallest of the centered slope and twice the one-sided slopes or zero if the one-sided slopes are of opposite sign.

The slope reconstruction only increases the spatial accuracy of the numerical fluxes. To increase the temporal accuracy, one must better approximate the solution to the initial problem with piecewise linear initial data, as opposed to piecewise constant initial data. Hancock [22] devised a simpler predictor-corrector approach that is used often in practice.

Dimension-by-dimension, we advance the solution in each cell over a half time-step using the primitive equations, that is $\partial_t \mathbf{w} + \mathbf{A}(\mathbf{w})\partial_x \mathbf{w} = 0$. Piecewise linear reconstruction on these predicted cell-centered values produces approximate predicted values at the cell interfaces:

$$\mathbf{w}_{i\pm 1/2,j,k,\mp}^{n+1/2} = \mathbf{w}_{ijk}^n - \left[\frac{\Delta t}{2} \mathbf{A}(\mathbf{w}_{ijk}) \mp \frac{\Delta x}{2} \mathbf{I} \right] (\boldsymbol{\sigma}_{i,j,k}^x)^n, \quad (14)$$

where $(+/-)$ corresponds to the right/left of the interface. Note that it is sufficient to use the old slopes instead of recomputing them after the predictor step. Predicted interface values (14), used as inputs to the Riemann solver, produce high-resolution interface fluxes.

4.3. Numerical Results

A simple multifluid result of an acoustic pulse in a uniform flow is shown in Figure 4. Here, an initial spike in T_e and p in an Au plasma drives an acoustic wave, which is transmitted with little reflection through the material interface (near $z = 680\lambda$) into a CH plasma with a faster sound speed. Note that the total pressure remains constant across the moving material interface.

Results of a recent NIF-relevant calculation made possible by the multifluid hydrodynamics solver are shown in Figure 5. The initial data are taken from a radiation hydrodynamics simulation of an indirect-drive target (hohlraum) near the wall, where a region of Au plasma propagates into an He plasma. The incident laser generates ion acoustic waves that reflect the light in a process known as stimulated Brillouin scattering. Most of the backscatter occurs in the vicinity of the material interface where the volume fraction of Au is over 0.5. Note that the irregular material boundary is an artifact of the zoning in the radiation hydrodynamics data.

5. Conclusions

Working with domain application scientists, we are involved in an on-going process to improve the algorithms in and, thus, the applicability of their parallel LPI code **pF3d**. We have developed an efficient, two-level, nonlinear electron heat conduction capability, making use of our knowledge of adaptive mesh refinement, FAC, and multigrid methods. In addition, through a

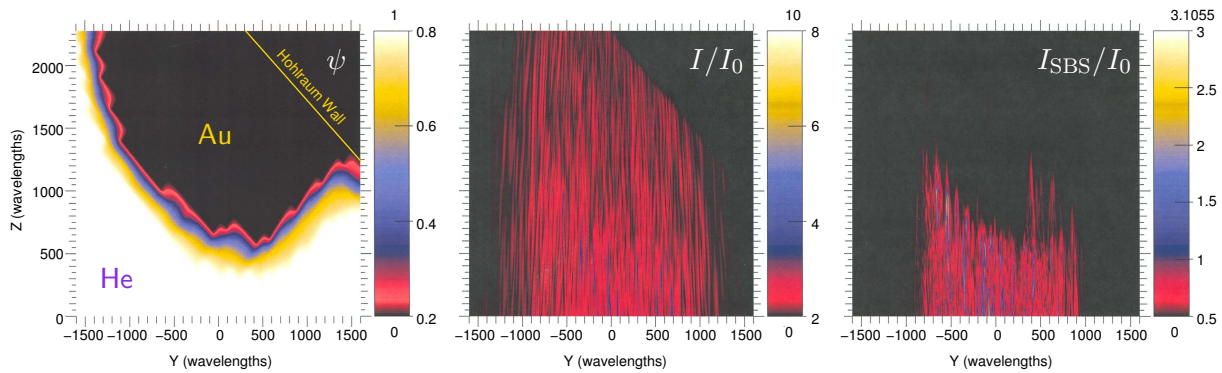


Figure 5. Multifluid hohlraum calculation corresponding to Au and He plasmas. Volume fraction ψ , laser intensity I/I_0 ($I_0 = 3(10^{14})$ W/cm²), and stimulated Brillouin backscatter I_{SBS}/I_0 at $t = 76.67$ ps are plotted for a domain of size $320\lambda \times 3200\lambda \times 2048\lambda$ ($\lambda = 0.351\mu\text{m}$), using a resolution of $128 \times 1280 \times 768$. The simulation was run on 2048 processors of BlueGene/L.

careful structuring of the governing equations, we have constructed a robust, efficient, nonlinear multifluid algorithm based on high-resolution Godunov methods. Both technologies are allowing physicists to study problems of practical relevance for the ICF efforts at LLNL. Nevertheless, due to the wide range of spatial and temporal scales, computational LPI is still in need of major algorithmic improvements, in order to simulate routinely at NIF-relevant engineering scales.

Acknowledgments

We wish to thank our physics collaborators C. Still, R. Berger, L. Divol, B. Langdon, and E. Williams, and we thank the NIF Programs for funding our efforts. This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48. UCRL-CONF-222175

References

- [1] Lindl J D *et al* 2003 *Phys. Plasmas* **11** 339-491
- [2] Still C H *et al* 2000 *Phys. Plasmas* **7** 2023-32
- [3] Dorr M R, Garaizar F X and Hittinger J A F 2002 *J. Comput. Phys.* **177** 233-63
- [4] Hittinger J A F, Dorr M R, Berger R L and Williams E A 2005 *J. Comput. Phys.* **209** 695-729
- [5] Berger R L *et al* 1998 *Phys. Plasmas* **5** 4337-56
- [6] Feit M D and Fleck J A 1988 *J. Opt. Soc. Am. B* **5** 633-40
- [7] Epperlein E 1990 *Phys. Rev. Lett.* **65** 2145-8
- [8] Spitzer L and Härm R 1953 *Phys. Rev.* **89** 977-81
- [9] McCormick S F and Thomas J 1986 *Math. Comput.* **46** 439-56
- [10] Trottenberg U, Schüller A and Oosterlee C 2000 *Multigrid* (New York: Academic Press)
- [11] Wesseling P 1996 *An Introduction to Multigrid Methods* (New York: John Wiley & Sons)
- [12] Abgrall R and Karni S 2001 *J. Comput. Phys.* **169** 594-623
- [13] Colella P 1990 *J. Comput. Phys.* **87** 171-200
- [14] Saltzman J 1994 *J. Comput. Phys.* **115** 153-68
- [15] Strang G 1968 *SIAM J. Numer. Anal.* **5** 506-17
- [16] Godunov S K 1959 *Math. Sbornik* **47** 271-306
- [17] Harten A, Lax P D and Van Leer B *SIAM Rev.* **25** 35-61.
- [18] Toro E F 1999 *Riemann Solvers and Numerical Methods for Fluid Dynamics* (New York: Springer-Verlag)
- [19] Einfeldt B *et al* 1991 *J. Comput. Phys.* **92** 273-95
- [20] Batten P *et al* 1997 *SIAM J. Sci. Comput.* **18** 1553-70
- [21] Van Leer B 1979 *J. Comput. Phys.* **32** 101-36
- [22] Van Leer B 1984 *SIAM J. Sci. Stat. Comput.* **5** 1-20